

Copyright
by
Lihao Zhang
2015

The Report Committee for Lihao Zhang
Certifies that this is the approved version of the following report:

Statistical Clustering of Data

APPROVED BY

SUPERVISING COMMITTEE:

Thomas W. Sager, Supervisor

Matthew Hersh

Statistical Clustering of Data

by

Lihao Zhang, B.S.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN STATISTICS

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2015

Dedicated to my loving parents.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Thomas W. Sager, whose constant support and constructive advice were always encouraging me to learn more and improve a lot. This report could not be accomplished without his dedication. In the mean time, I wish to express my sincere thanks to Dr. Matthew Hersh, reader of this report, for his patience in providing comments and assistance.

Finally, I wish to express my special thanks to my parents for their unconditional love and support every time, especially the hardest times.

Statistical Clustering of Data

Lihao Zhang, M.S. Stat
The University of Texas at Austin, 2015

Supervisor: Thomas W. Sager

Cluster analysis aims at segmenting objects into groups with similar members and, therefore helps to discover distribution of properties and correlations in large datasets. Data clustering has been widely studied as it arises in many domains in marketing, engineering, and social sciences. Especially, the occurrence of transactional and experimental datasets in large scale in recent years significantly increased the necessity of clustering techniques to reduce the size of the existing objects, to achieve a better knowledge of the data.

This report introduced fundamental concepts related to cluster analysis, addressed the similarity and dissimilarity measurements for cluster definition, and clarified three major clustering algorithms-hierarchical clustering, K -means clustering and Gaussian mixture model fitted by Expectation-Maximization (EM) algorithm-theoretically and experimentally to illustrate the process of clustering. Finally, methods of determining the number of clusters and validating the clustering were presented as for clustering evaluation.

Table of Contents

Acknowledgments	v
Abstract	vi
List of Tables	ix
List of Figures	x
Chapter 1. Introduction	1
Chapter 2. Data Clustering	3
2.1 Unsupervised Learning	4
2.1.1 Definition of a Cluster	5
2.1.2 General Steps for Cluster Analysis	7
2.2 Similarities and Dissimilarities	7
2.2.1 Correlation Coefficients	8
2.2.2 Distance Measurements	9
Chapter 3. Clustering Algorithms	11
3.1 Connectivity-based Model	11
3.1.1 Hierarchical Agglomerative Clustering	11
3.1.2 Clustering Analysis on Romano-British Pottery	12
3.2 Centroid-based Model	16
3.2.1 K -means Clustering	16
3.2.2 K -means Experimental Study on Pottery Data	17
3.2.3 Extensions of K -means	19
3.3 Model-based Clustering	20
3.3.1 Gaussian Finite Mixture Models	21
3.3.2 Expectation-Maximizations Clustering	25
3.3.3 Experimental Analysis on iris Dataset	27

Chapter 4. Cluster Validation	32
4.1 Determination of Number of Clusters	32
4.1.1 Invalidity of Statistical Significance Testing	33
4.1.2 The Elbow Method	33
4.1.3 Information Criterion Approach	35
4.1.4 Three Top Performing Heuristic Methods	35
4.2 Cluster Validation	36
Appendices	39
Appendix A. R Code for Figure 2.2 & Clustering Analysis on Romano-British Pottery	40
Appendix B. R Code for K-means Experimental Study on Pot- tery Data	42
Appendix C. R Code for Experimental Analysis on iris Dataset	44
Bibliography	45
Vita	49

List of Tables

3.1	Romano-British Pottery Data	13
3.2	Relations between Clusters and Kiln Sites for Average Link . .	16
3.3	Total Within-cluster Sum of Squared Distance	17
3.4	Cluster Means for Each Cluster	19
3.5	Relations between Clusters and Kiln Sites using K -means . . .	19
3.6	Brief Results of EM Clustering	29
3.7	Parameter Estimates of Mixing Probabilities and Means . . .	29
3.8	Parameter Estimates of Covariances	30
4.1	Percentage of Explained Variance against Number of Clusters	35

List of Figures

2.1	Clusters with Diversity	3
2.2	Clusters for iris Dataset	6
3.1	Image of Euclidean Distance based Dissimilarity Matrix on Pot- tery Data	14
3.2	Dendrogram of Hierarchical Clustering using Euclidean Distance	15
3.3	The Way K -means Clustering Works	17
3.4	Relations between SSD and Number of Clusters	18
3.5	Gaussian Mixture Model with Two Gaussian Distributions . .	23
3.6	The Bivariate iris Dataset	28
3.7	Density Estimate for Bivariate iris Dataset	28
3.8	Plots Associated with the Function Mclust for iris dataset . .	30
4.1	Explained Variance by Clustering against Number of Clusters	34

Chapter 1

Introduction

Cluster analysis or clustering, also called data segmentation, is related to grouping or segmenting a set of objects such that objects in the same group (called a cluster) are to some extent similar to each other, while are dissimilar (in some sense) to objects belong to other groups. It is an unsupervised learning method in data mining, and a common technique for statistical data analysis used in various areas, including marketing, psychology, linguistics, bioinformatics, machine learning, pattern recognition, etc.

Cluster analysis can also be used as a way to generate descriptive statistics or visual aids to determine the potential existence of a set of distinct subgroups within a sample dataset, each subgroup representing objects with significantly different properties. Realization of that objective requires evaluation of the differences between the objects assigned to their respective clusters.

Two major problems that cluster analysis concerns are to determine the number of clusters and assign objects into each cluster appropriately. A variety of clustering algorithms have been set up in order to effectively solve such problems.

Cluster analysis [1] was originated in anthropology by Driver and Kroe-

ber in 1932 and introduced to psychology by Zubin in 1938 and Robert C. Tryon in 1939 [2], and famously used by Raymond B. Cattell beginning in 1943 [3] for trait theory classification in personality psychology.

Generally, data analysis mainly involves predictive modeling: given a set of training data, we want to predict the class memberships of a set of test data using predictive models based on training data. This kind of task is also called learning. Learning problems are typically classified into two categories, supervised learning (mainly classification) and unsupervised learning (mainly clustering). Supervised learning deals with labeled data, while unsupervised learning involves unlabeled data [4]. In this way, clustering is more abstract and challenging than classification. More recently, semi-supervised learning [5] has been proposed. It specified pairwise constraints (a "weaker" way of specifying the prior knowledge of the wanted model) instead of labeling the class, and constraints are thought to be advantageous to data clustering [6] [7] [8].

The purpose of this report is to to illustrate the process of cluster analysis, specifically, to clarify three major clustering algorithms-hierarchical clustering, K -means clustering and Gaussian mixture model fitted by Expectation-Maximization (EM) algorithm-theoretically and experimentally on real world datasets. In clustering algorithms, this report emphasizes on the statistical basis of Gaussian mixture model approach by EM algorithm, associated with its experimental analysis based on the **iris** dataset.

Chapter 2

Data Clustering

The objective of data clustering, also known as cluster analysis, is to discover the natural groupings of a set of data points, patterns or objects, including determining the number of groups and assigning certain individuals into respective groups. An example [9] of clustering is shown in Figure 2.1.

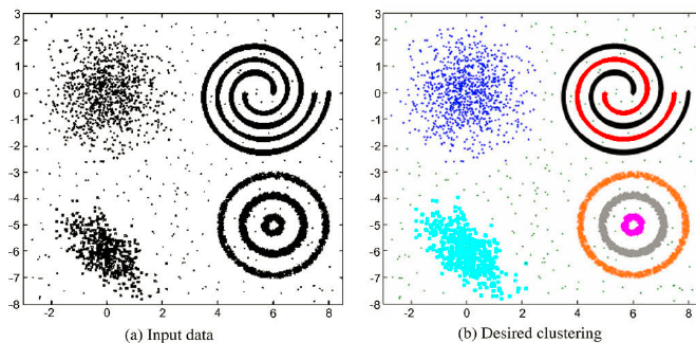


Figure 2.1: Clusters with Diversity

In Figure 2.1 (a), the input data is unlabeled and all the data points, including the scattered dots, circles and swirls, pertain to one large group. After a certain algorithm has been implemented, seven clusters are generated and distinguished by colors in Figure 2.1 (b) with dissimilarities in shape, size and density.

2.1 Unsupervised Learning

Clustering is a typical unsupervised learning technique. To establish a context, supervised learning will be firstly explained.

Supervised learning is a data mining task of inferring a function from labeled training data. Suppose we observe a response variable Y and m different predictor variables x_1, x_2, \dots, x_m , and if there is a potential relationship between the response Y and the predictors $X = (x_1, x_2, \dots, x_m)$, such relation could be presented in the following general form:

$$E(Y) = F(X)$$

In the function above, $E(Y)$ is the expected value of the response Y , F indicates the potential but unknown relation.

Under many situations, predictors are readily available, but Y often is hard to get. Then Y could be predicted by:

$$\hat{Y} = \hat{F}(X)$$

where \hat{Y} is the prediction of the response variable, and \hat{F} is the estimate of the relation between the response and the predictors.

In supervised learning, the goal [10] is to fit a model that relates the response variable to the predictor variables, aiming at accurately predicting the response for future observations (prediction) or better understanding the relationship between the response and the predictors (inference).

In contrast, unsupervised learning describes a more challenging situation in which for each observation $i = 1, \dots, n$, we have a set of measurements x_i but no associated response y . It is unable to fit a predictive model (regression, decision tree, support vector machine, k-nearest neighbor, etc), since observations are unlabeled, and there is no response variable to predict. Such situation is referred to as *unsupervised*, as a response variable is lacking to supervise the learning.

After classifying learning problems as supervised and unsupervised, semi-supervised learning was proposed, in which a small portion of the training data is labeled, and the unlabeled data, instead of being dropped, are also used during the learning process.

2.1.1 Definition of a Cluster

Everitt [11] has studied the cluster and given a specific definition of a cluster by evaluating the circular nature of most proposed definitions. A cluster is a continuous region of variable space containing a relatively high density of data points separated from other high density regions by areas containing a relatively low density of data points.

In clustering problems, things to be clustered are usually called objects or observations, also known as patterns, cases or entities. The aspects of these objects for evaluating their similarities or dissimilarities are often and variously called variables, attributes, features, or characters.

An example of four clusters is shown in Figure 2.2 by using the **iris**

flower dataset, a classical dataset from R. A. Fisher. The **iris** consists of 150 observations with 4 numerical attributes sepal length, sepal width, petal length and petal width, and 1 categorical attribute, species, with 3 categories setosa, versicolor and virginica. In Figure 2.2, after neglecting the categorical attribute, all the data points are grouped into four distinct clusters based on two principal components of numerical characters in the original dataset.

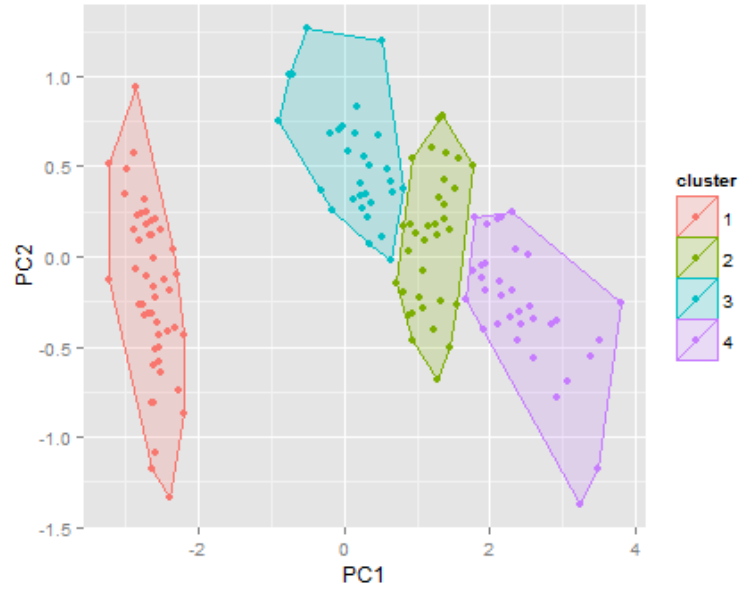


Figure 2.2: Clusters for **iris** Dataset

There are several characteristics of clusters which could be quantified when observations are plotted as points in a p -dimensional variable space:

- (1) Density – the within-cluster points are highly concentrated
- (2) Variance – the dispersion from the cluster centroid is small for the points in the cluster

- (3) Dimension – the size or radius of the cluster is small
- (4) Shape – clusters are typically ellipsoidal
- (5) Separation – clusters rarely overlap or they are disconnected.

Most of the clustering algorithms are set up based on these characteristics.

2.1.2 General Steps for Cluster Analysis

- *Feature Selection*: select observations to be clustered and define attributes used for clustering such observations
- *Proximity Measure*: compute similarities or dissimilarities among observations
- *Clustering Criterion*: express via a cost function or certain rule and choose proper clustering algorithm
- *Cluster Generation*: create clusters of similar objects
- *Cluster Validation*: validate the resulting clusters and interpret the result

2.2 Similarities and Dissimilarities

In a data matrix, observations could be thought of as the rows, with attributes as the columns. If the number of observations and attributes are n and p , respectively, the size of the data matrix would be $n \times p$.

Then the proximity matrix between pairs of observations could be computed in terms of the data matrix, it could be either similarity matrix or dis-

similarity matrix, and typically, correlation is used for computing similarity, while distance is used to measure dissimilarity.

2.2.1 Correlation Coefficients

For the similarity matrix resulting from the data matrix $n \times p$, the entry in row i and column j of the similarity matrix shows how similar (dissimilar) object i and object j are. In this way, the similarity matrix must be $n \times n$.

If correlation is used, the resulting similarity matrix is very different from the ordinary correlation matrix for the same $n \times p$ data matrix. In this case, the ordinary correlation matrix would be of size $p \times p$. The entry in row i and column j of the ordinary correlation matrix shows how similar variable i and variable j are in the data matrix. On the other hand, the entry in row i and column j of a similarity matrix shows how similar observation i and observation j are. That is to say, an $n \times n$ similarity matrix shows how similar the rows in the $n \times p$ data matrix are, whereas the ordinary $p \times p$ correlation matrix shows how similar the columns in the $n \times p$ data matrix are.

So if one observation $X_i = \{x_{i1}, \dots, x_{ip}\}$ containing p values and another observation $X_j = \{x_{j1}, \dots, x_{jp}\}$ containing p values, and $\forall i, j = 1, \dots, n, i \neq j$, then the sample correlation coefficient for the similarity matrix would be:

$$r = r_{X_i X_j} = \frac{\sum_{k=1}^p (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{\sum_{k=1}^p (x_{ik} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^p (x_{jk} - \bar{x}_j)^2}}$$

Correlation measures are rarely used in practice for computing similarities.

2.2.2 Distance Measurements

Most algorithms [12] presume a matrix of dissimilarities with non-negative entries and zero diagonal elements: $d_{ii} = 0, i = 1, 2, \dots, n$. If the original data were collected as similarities, a suitable monotone-decreasing function can be used to convert them to dissimilarities. Also, most algorithms assume symmetric dissimilarity matrices, so if the original matrix D is not symmetric, it could turn to be symmetric by replacing it with $(D + D^T)/2$. Subjectively constructed dissimilarities are seldom distances in the strict sense, since the triangle inequality $d_{ii'} \leq d_{ik} + d_{ki'}$, for all $k \in \{1, \dots, n\}$ does not hold. Thus, some algorithms that assume distances cannot be used with such data.

Generally, for $\forall i, j, k \in 1, 2, \dots, n$, and $i \neq j \neq k$, the dissimilarity metric such that:

$$d(i, j) \geq 0$$

$$d(i, i) = 0$$

$$d(i, j) = d(j, i)$$

$$d(i, j) \leq d(i, k) + d(k, j)$$

And common distance measures for dissimilarities between observations are:

$$(1) \text{ Euclidean Distance} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2}$$

$$(2) \text{ Manhattan Distance} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

$$(3) \text{ Mahalanobis Distance} = (\mathbf{X}_i - \mathbf{X}_j)' \mathbf{S}^{-1} (\mathbf{X}_i - \mathbf{X}_j)$$

where x_{ik} is the value of observation i on variable k , x_{jk} is the value of observation j on variable k , $(\mathbf{X}_i - \mathbf{X}_j)'$ is the $1 \times p$ row vector of differences between row i and row j of the data matrix, and \mathbf{S}^{-1} is the inverse of the $p \times p$ covariance matrix of the attributes of data. And it is standard to standardize variables prior to calculating distance measures used as dissimilarities.

Chapter 3

Clustering Algorithms

3.1 Connectivity-based Model

Connectivity based clustering, also known as hierarchical clustering, is based on the basic idea of general clustering thoughts that objects are more similar to nearby objects than to objects farther away.

Hierarchical clustering methods require the user to specify a measure of dissimilarity (mostly are the three distance measures introduced in 2.2.2) between disconnected groups of observations. As the name suggests, they produce hierarchical representations, called a *dendrogram*, in which the groups at each level of the hierarchy are generated by merging two lower-level groups. In this way, at the lowest level in this hierarchy, each group contains one single observation; and at the highest level, there is only one group which contains all the data information.

3.1.1 Hierarchical Agglomerative Clustering

There are two basic paradigms for hierarchical clustering: *agglomerative* and *divisive*. The former paradigm will give a bottom-up dendrogram, while the latter one is expected to show a top-down dendrogram. Four major

methods are used to compute the similarity of clusters where each of them may contain multiple instances:

- *Single Link*: Nearest neighbor, clustering two most similar members.
- *Complete Link*: Farthest neighbor, clustering two least similar members.
- *Average Link*: Average neighbor, clustering members in average similarity.
- *Ward's Method*: Minimum variance criterion, clustering objects by minimizing the total within-cluster variance.

The basic procedures for all of the three methods are similar:

Step 1 Start with clusters C_1, C_2, \dots, C_n , each contains a single observation.

Step 2 Find the proper pair of distinct clusters, say C_i and C_j , merge C_i and C_j , and decrease the number of clusters by 1.

Step 3 When # cluster equals to 1, stop the process; else, return to Step 2.

3.1.2 Clustering Analysis on Romano-British Pottery

This section clarifies how hierarchical agglomerative clustering works by using the data of Romano-British Pottery [13].

The data shown in Table 3.1 conveys numeric information for the chemical composition of Romano-British pottery with 45 specimens, determined by atomic absorption spectrophotometry [14] with the values for nine oxides [15].

In addition to the chemical composition of the pots, the kiln site at which the pottery was found is categorized into 5 places, with label 1 through 5. Based on such group of data, people would like to know whether, in terms of these chemical compositions, the pots can be divided into distinct clusters, and how these groups are related to the kiln site.

Table 3.1: Romano-British Pottery Data

Al_2O_3	Fe_2O_3	MgO	CaO	Na_2O	K_2O	TiO_2	MnO	BaO	Kiln
18.8	9.52	2.00	0.79	0.40	3.20	1.01	0.077	0.015	1
16.9	7.33	1.65	0.84	0.40	3.05	0.99	0.067	0.018	1
18.2	7.64	1.82	0.77	0.40	3.07	0.98	0.087	0.014	1
...
16.7	0.92	0.53	0.01	0.05	1.76	0.91	0.004	0.013	5
14.8	2.74	0.67	0.03	0.05	2.15	1.34	0.003	0.015	5
19.1	1.64	0.60	0.10	0.03	1.75	1.04	0.007	0.018	5

Among the 45 observations of pottery data, first 21 observations belong to kiln site 1, next 12 observations belong to kiln site 2, next 2 observations kiln site 3, next 5 observations kiln site 4, and the last 5 observations belong to kiln site 5.

For hierarchical clustering analysis, first, an intuitive impression of the potential clusters of all the 45 specimens of chemical composition of pottery is necessary. In Figure 3.1, the image of dissimilarity matrix of the pottery data using Euclidean distance is given. Each cell of the dissimilarity matrix in the plot has a color-based value from 0 through 12 with color from pink across turquoise, which means the closer to pink the color is, the closer to zero Euclidean distance the cell will be, indicating such group of cells more probably

belong to the same cluster. Specifically, Figure 3.1 leads to a direct impression that at least 3 clusters exist with much smaller within-cluster distances than those can be seen in other cells.

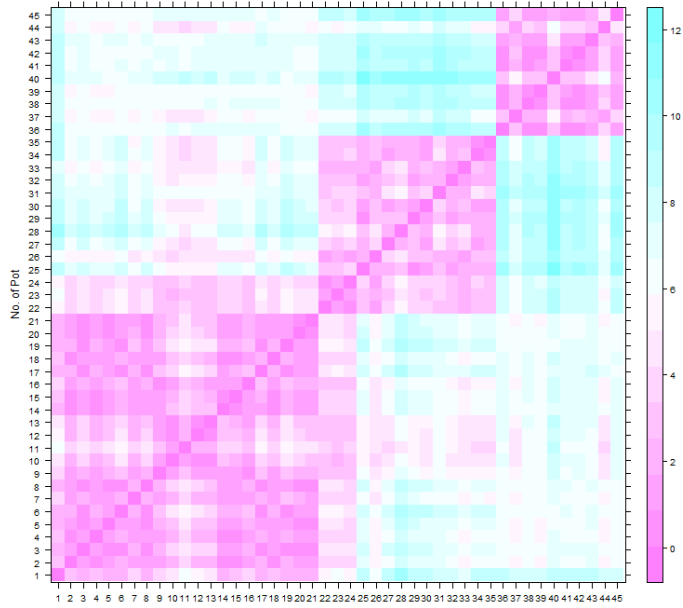


Figure 3.1: Image of Euclidean Distance based Dissimilarity Matrix on Pottery Data

Then, hierarchical agglomerative algorithms is implemented using the first three methods in computing similarities, with dendrograms to visualize the results of hierarchical clustering. This could be realized by the R function **hclust**, a function specialized in hierarchical cluster analysis. As is shown in Figure 3.2, each dendrogram of single link, complete link and average link, has 3 clusters.

Typically, to determine the number of clusters for a dendrogram, one

needs to look for natural groupings defined by long stems. If we cut at height = 2.0 for single link, height = 7.0 for complete link and height = 4.0 for average link, we get exactly 3 clusters in each of the three dendrograms. If Manhattan distance or Mahalanobis distance is used as measurement of dissimilarity, then the dendrograms would be a little bit different.

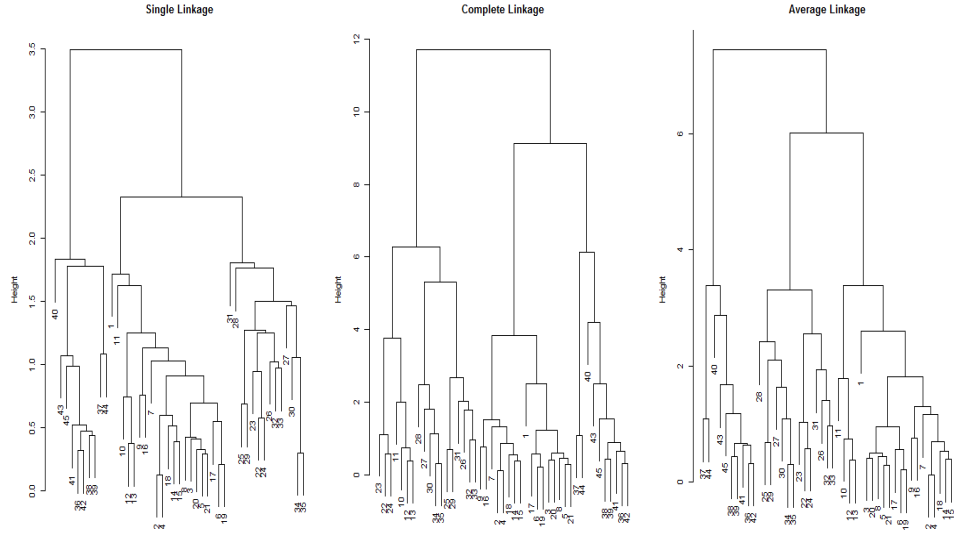


Figure 3.2: Dendrogram of Hierarchical Clustering using Euclidean Distance

In such case, the problem of whether the pots can be divided into distinct clusters has been solved, then the problem of how these clustering relate to kiln sites could also be solved. The relationships between the clusters and the kiln sites could be seen in Table 3.2. Choose the average link as an example, all the 21 pots from Kiln Site 1 belong to Cluster 1; all the 12 pots from Kiln Site 2 and all the 2 pots from Kiln Site 3 are found in Cluster 2; all the pots from Kiln Site 4 and 5 are found in Cluster 3.

Table 3.2: Relations between Clusters and Kiln Sites for Average Link

# Cluster	Kiln 1	Kiln 2	Kiln 3	Kiln 4	Kiln 5
1	21	0	0	0	0
2	0	12	2	0	0
3	0	0	0	5	5

3.2 Centroid-based Model

In centroid-based clustering, clusters are represented by a central vector, usually called a *centroid*, which may not necessarily be an existing member of the current data set. When the number of clusters k is fixed, K -means clustering tries to solve such an optimization problem: find the optimal k centroids of potential clusters and assign all the objects to proper centroids in creating clusters, such that the total sum of squared distances between each object and its centroid within a cluster are minimized.

3.2.1 K -means Clustering

Typically, the procedure for K -means clustering would be:

Step 1 Randomly pick data points as initial representatives.

Step 2 Assign each data point to its closest representative.

Step 3 Recompute "means" for each potential cluster.

In Figure 3.3, an intuitive impression for the procedure of K -means clustering is shown from left to right. And such procedure is constrained to minimize the total sum of squared distances between data points and the centroid within each cluster, the objective function is given below:

$$J = \operatorname{argmin}_{\{\mu_1, \dots, \mu_k\}} \sum_{c=1}^k \sum_{i=1, x_i \in \chi_c} \|x_i - \mu_c\|^2$$

where μ_1, \dots, μ_k are k centroids, which are the means within each cluster, $x_i \in \chi_c$ indicates the i^{th} data point in the c^{th} cluster, $c = 1, \dots, k$.

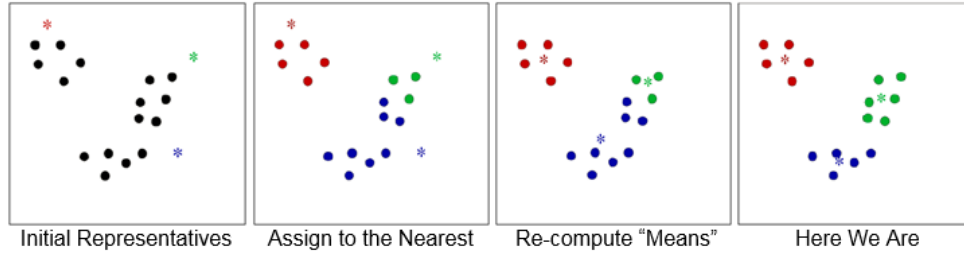


Figure 3.3: The Way K -means Clustering Works

3.2.2 K -means Experimental Study on Pottery Data

This section clarifies how K -means clustering works using the same pottery data used in Section 3.1.2. The pottery data is grouped into 1 through 10 clusters, respectively, the values of "total within-cluster sum of squared distance" for 1 cluster through 10 clusters are listed in Table 3.3.

Table 3.3: Total Within-cluster Sum of Squared Distance

# C	1	2	3	4	5	6	7	8	9	10
SSD	753.7	402.8	145.1	116.8	94.5	74.1	57.6	55.0	55.6	34.1

It can be found in Table 3.3 that the total within-cluster SSD decreases when the data is grouped into 1 cluster through 8 clusters, with the value

decreases from 753.7 to 55.0. Then the value for such measurement increases a little to 55.6 when there are 9 clusters and drops to 34.1 when there are 10 clusters.

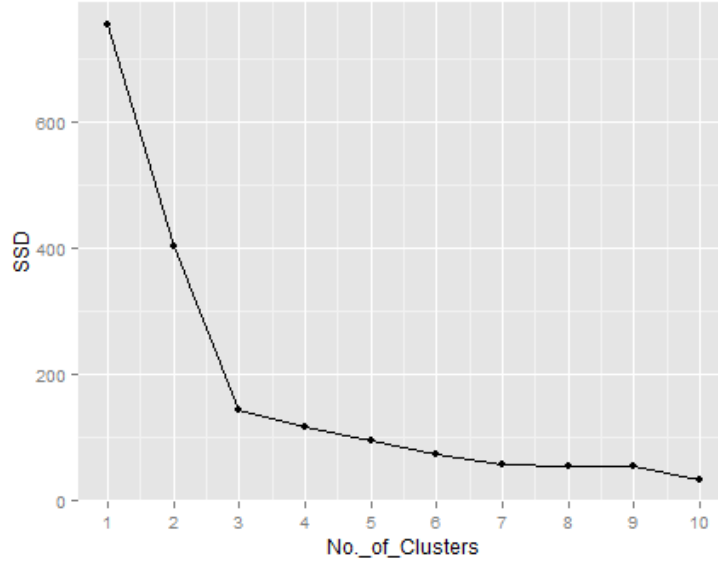


Figure 3.4: Relations between SSD and Number of Clusters

Values in Table 3.3 could be reflected in Figure 3.4, and there is a decreasing trend between number of clusters and the total within-cluster SSD. By means of the "Elbow" method for determining the number of clusters, the optimal number of clusters using K -means clustering would be three, which matches the number when we use hierarchical agglomerative clustering method based on the same dataset. And when 3 clusters are generated, the explained variance of the clusters (between-cluster SSD) is 80.8 % of the total variance, which is good enough.

The cluster means in terms of each chemical composition could be found in Table 3.4.

Table 3.4: Cluster Means for Each Cluster

# C	Al ₂ O ₃	Fe ₂ O ₃	MgO	CaO	Na ₂ O	K ₂ O	TiO ₂	MnO	BaO
1	12.44	6.21	4.78	0.21	0.23	4.19	0.68	0.12	0.02
2	17.75	1.61	0.64	0.04	0.05	2.02	1.02	0.03	0.02
3	16.92	7.43	1.84	0.94	0.35	3.10	0.94	0.07	0.02

Meanwhile, a table is available to show the relations between clusters and kiln sites in Table 3.5. All the 12 pots from Kiln Site 2 and all the 2 pots from Kiln Site 3 are found in Cluster 1, all the pots from Kiln Site 4 and 5 are found in Cluster 2, and all the 21 pots from Kiln Site 1 belong to Cluster 3.

Table 3.5: Relations between Clusters and Kiln Sites using *K*-means

# Cluster	Kiln 1	Kiln 2	Kiln 3	Kiln 4	Kiln 5
1	0	12	2	0	0
2	0	0	0	5	5
3	21	0	0	0	0

In such case, it is clear that relations between clusters and kiln sites resulted from *K*-means clustering in Table 3.5 and those resulted from average link of hierarchical clustering in Table 3.2 are the same, despite the difference in the notation of cluster numbers.

3.2.3 Extensions of K-means

Based on the basic algorithms of *K*-means clustering, many extensions have been shown in public. Some of these extensions involves with merging,

splitting clusters and minimizing the cluster size. Two famous derivation of K -means in pattern recognition are the ISODATA method by Ball and Hall [16] and the FORGY method by E.W. Forgy [17].

Besides, there's one significant fact in K -means is that, each data element is finally assigned to one distinct cluster, clustering in such way is also called *hard clustering*. One important extension of K -means clustering is called *Fuzzy c-means*, proposed by J.C. Dunn [18] and improved by J.C. Bezdek [19], indicating that each data point can belong (with different amounts of membership level) to multiple cluster. Fuzzy clustering technique is also referred to as *soft clustering*, compared to hard clustering.

3.3 Model-based Clustering

Model-based clustering, also known as distribution-based clustering, is a general clustering technique based on distribution models. This clustering method defines clusters as observations belonging most likely to the same distribution, and it is similar to the way that some sample data set are generated by Gibbs sampling or MetropolisHastings in Bayesian statistics, by sampling observations from a distribution.

Actually, distribution-based clustering often suffers from a common problem which is overfitting, if there are no constraints set up for the model complexity. In such case, a model with higher complexity for the clustering job would better explain the data, which, at the mean time, add up to the inherent difficulty of choosing the appropriate complexity of the model.

One effective and commonly used method to solve the problem above is known as Gaussian mixture models, which is often fitted by expectation-maximization algorithm. Specifically, the data set or objects are usually modeled with a finite (to avoid overfitting) number of normal distributions, such normal distributions are randomly initialized, and then the initial parameters of such distributions are iterated with multiple runs to reach optimum in order to better fit the data set or objects.

3.3.1 Gaussian Finite Mixture Models

Gaussian finite mixture model is a linear combination of finite number of Gaussian distributions. Several necessary terms related to such model are clarified step by step.

- *Gaussian Distribution*

Let X be a normally distributed random variable with mean μ and standard deviation σ , for $\forall x_i \in X, i = 1, 2, \dots, n$, the probability function of X would be:

$$f(x_i) = N(x_i|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2\right\}$$

- *Likelihood Function*

Then the likelihood function $L(\mu, \sigma|x_i)$ of μ and σ for the given x_1, x_2, \dots, x_n would be:

$$L(\mu, \sigma|x_i) = \prod_{i=1}^n N(x_i|\mu, \sigma^2)$$

- *Multivariate Gaussian and Log-likelihood*

Let $\mathbf{x} = \{X_1, X_2, \dots, X_n\}$ be a n -dimensional random vector, if every linear combination of its n components has a univariate normal distribution, then \mathbf{X} is said to be n -variate normally distributed, with the multivariate Gaussian distribution:

$$\mathbf{x} \sim \mathcal{N}_n(\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right\}$$

where μ is the n -dimension mean vector with $\mu = [E[X_1], E[X_2], \dots, E[X_n]]$, and Σ is the $n \times n$ covariance matrix with $\Sigma = [\text{Cov}[X_i, X_j]], i = 1, 2, \dots, n; j = 1, 2, \dots, n$.

Then the log-likelihood function would be:

$$\begin{aligned} l(\mu, \Sigma | X_i) &= \ln \prod_{i=1}^n N(X_i | \mu, \Sigma) \\ &= -\frac{n}{2} \ln |\Sigma| - \frac{1}{2} \sum_{i=1}^n (X_i - \mu)^T \Sigma^{-1} (X_i - \mu) + \text{constant} \end{aligned} \tag{3.1}$$

Thus, the Maximum Likelihood Estimator of μ and Σ would easily be found:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n X_i$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T$$

- *Gaussian Finite Mixture Models*

Gaussian finite mixture models is a weighted sum of finite number of Gaussian probability density functions. Suppose we have K multivariate Gaussian distributed random vectors $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K$, μ_k and Σ_k are each vector's mean vector and covariance matrix, then the probability density function for the Gaussian finite mixture model would be:

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k N(\mathbf{x}|\mu_k, \Sigma_k)$$

where π_k is the weight with $\sum_{k=1}^K \pi_k = 1$, $0 \leq \pi_k \leq 1$, and π_k , μ_k and Σ_k are parameters to be estimated. $N(\mathbf{x}|\mu_k, \Sigma_k)$ is called the k^{th} component model, each component is a multidimensional Gaussian with its own mean μ_k and covariance matrix Σ_k . Figure 3.5 [20] is how $p(\mathbf{x})$ would be like and changing associated with the change of its parameters.

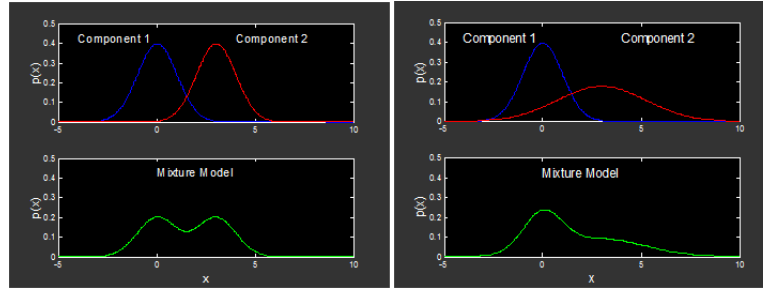


Figure 3.5: Gaussian Mixture Model with Two Gaussian Distributions

Clustering technique based on Gaussian Finite Mixture Models is used to cluster data points generated from each component model. The following two steps are to generate sample data points from each component model, and assign a latent variable to each of the data point.

Step 1 To generate data points

Firstly, randomly pick one of the components with probability π_k , then draw a sample data points $x_i, i = 1, 2, \dots, n$ from that component distribution

Step 2 Assign a latent variable to each data point

After each data point is generated from one of the K components, a latent variable $z_i = (z_{i1}, z_{i2}, \dots, z_{iK})$ is associated with each x_i , where $\sum_{k=1}^K$ and $p(z_{iK} = 1) = \pi_k$

Since the probability density function of Gaussian finite mixture model is already given, then the log-likelihood function for the generated data points from the Gaussian mixture is not hard to find:

$$\begin{aligned} l(\pi, \mu, \Sigma | \mathbf{x}) &= \ln L(\pi, \mu, \Sigma | \mathbf{x}) = \ln \prod_{i=1}^n \sum_{k=1}^K \pi_k N(\mathbf{x}_i | \mu_k, \Sigma_k) \\ &= \sum_{i=1}^n \ln \left\{ \sum_{k=1}^K \pi_k N(\mathbf{x}_i | \mu_k, \Sigma_k) \right\} \end{aligned} \tag{3.2}$$

Right now, values of the latent variables, which are associated with data points x_i , are not known, then the maximization of the incomplete log likelihood is needed, and Expectation-Maximization (EM) algorithm can be used to estimate the mixture parameters π_k , μ_k and Σ_k by iteratively maximizing the likelihood.

3.3.2 Expectation-Maximizations Clustering

The ExpectationMaximization (EM) algorithm is an iterative method for finding estimates of parameters by maximizing the log-likelihood in statistical models, which depend on unobserved latent variables. The EM iteration alternates between performing an expectation step (E-step) and a maximization step (M-step), where the E-step creates a function for the expectation of the log-likelihood which is evaluated by the current estimate for the parameters, and the M-step computes estimate of parameters by maximizing the expected log-likelihood resulted from the E-step. The parameter-estimates obtained from the M-step are then used to determine the distribution of the latent variables in the next E- step.

For the Gaussian Finite Mixture Model given in Section 3.3.1, the specific expectation and maximization steps are as follows.

- *E-step: evaluate "responsibilities" of each cluster of data points with the current parameters*

For the given initial parameters, it is doable to compute the expected values of the latent variables using Bayes Theorem, such values are also known as responsibilities of the data points.

$$\begin{aligned}
\tau_{ik} &= E(z_{ik}) = p(z_{ik} = 1 | x_i, \pi, \mu, \Sigma) \\
&= \frac{p(z_{ik} = 1)p(x_i | z_{ik} = 1, \pi, \mu, \Sigma)}{\sum_{k=1}^K p(z_{ik} = 1)p(x_i | z_{ik} = 1, \pi, \mu, \Sigma)} \\
&= \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)}
\end{aligned} \tag{3.3}$$

where responsibilities $\tau_{ik} \in [0, 1]$ and $\sum_{k=1}^K \tau_{ik} = 1$ for all i .

- *M-step: re-estimate parameters using the existing "responsibilities"*

Use the responsibilities to maximize the expected *complete* log-likelihood to give parameter updates.

Combine Equation 3.2 and Equation 3.3, then the maximum likelihood estimator with responsibilities for the parameters could be found:

$$\begin{aligned}
\frac{\partial l(\pi, \mu, \Sigma | x)}{\partial \mu_k} &= \sum_{i=1}^n \frac{\pi_k N(x_i | \mu_k, \Sigma_k)}{\sum_{k=1}^K \pi_k N(x_i | \mu_k, \Sigma_k)} \Sigma_k^{-1} (x_i - \mu_k) = 0 \\
&= \sum_{i=1}^n \tau_{ik} \Sigma_k^{-1} (x_i - \mu_k) = 0
\end{aligned} \tag{3.4}$$

$$\mu_k = \frac{\sum_{i=1}^n \tau_{ik} x_i}{\sum_{i=1}^n \tau_{ik}}$$

Similarly, optimizations of π_k and Σ_k are as follows:

$$\pi_k = \frac{\sum_{i=1}^n \tau_{ik}}{n}$$

$$\Sigma_k = \frac{\sum_{i=1}^n \tau_{ik} (x_i - \mu_k)(x_i - \mu_k)^T}{\sum_{i=1}^n \tau_{ik}}$$

After the estimates for the parameters are obtained, iterate E-step and M-step until the log-likelihood of data does not increase any more-reach maximum of log-likelihood, in such case, parameters are iteratively optimized to fit the data set.

In order to obtain a hard clustering, objects or data points are often then assigned to the Gaussian distribution they most likely belong to, for soft clustering, this is not necessary.

Model-based clustering produces complex models for clusters that can capture correlation and dependence between attributes. However, there's one obvious limitation for users: for many real data sets, there may be no well-defined distribution model, that is to say, for example, Gaussian distributed is a strong assumption on the data.

3.3.3 Experimental Analysis on iris Dataset

As is used in Section 2.1.1 as an example of clusters, the **iris** dataset here is another good example to show how model-based clustering works, especially the Gaussian finite mixture model fitted by Expectation-Maximization clustering algorithm.

In this section, only the first two columns Sepal.Length and Sepal.Width from **iris** are used, since the data with two dimensions makes the analysis more friendly to visualize. The bivariate **iris** dataset is shown in Figure 3.6.

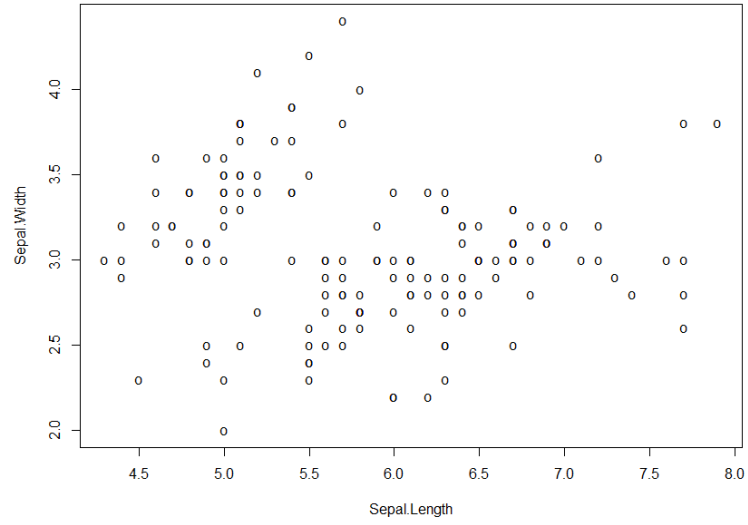


Figure 3.6: The Bivariate **iris** Dataset

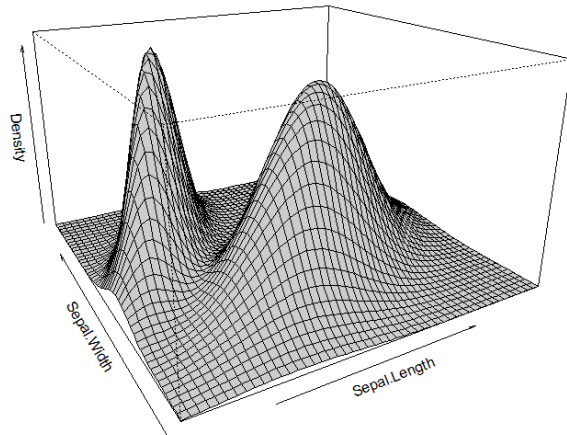


Figure 3.7: Density Estimate for Bivariate **iris** Dataset

And from the density estimate for the data in Figure 3.7, a relatively obvious mixture model with 2 Gaussian distributions can be seen, thus, the clustering technique Gaussian Finite Mixture Model fitted by EM algorithm is suitable to be operated on the data.

The R package **mclust** [21] is used, which is specialized in normal mixture modeling for model-based clustering, classification, and density estimation. The function **Mclust** performs clustering analysis based on Gaussian finite mixture model fitted by EM clustering, and a brief result is shown in Table 3.6, the corresponding plots are shown in Figure 3.8

Table 3.6: Brief Results of EM Clustering

Log-likelihood	Observations	DF	BIC	Clustering Table
-225.9263	150	10	-501.9589	#1: 49, #2: 101

In this case, the best model according to **BIC** is an variable-covariance model with 2 clusters, the maximum log-likelihood is -225.9263, optimal BIC - 501.9589, and 49 observations belong to Cluster 1 and 101 observations belong to Cluster 2.

Additionally, parameters π, μ, Σ in the Gaussian finite mixture model could be found by EM iterations which are shown in Table 3.7 and Table 3.8

Table 3.7: Parameter Estimates of Mixing Probabilities and Means

Cluster #	Mixing Probability π	Means μ
1	0.3223103	S.L: 5.016245; S.W: 3.454680
2	0.6776897	S.L: 6.236698; S.W: 2.868354

Table 3.8: Parameter Estimates of Covariances

Cluster # 1 Σ	Sepal.Length	Sepal.Width
Sepal.Length	0.12129045	0.09030555
Sepal.Width	0.09030555	0.11938505
Cluster # 2 Σ	Sepal.Length	Sepal.Width
Sepal.Length	0.4643624	0.1251138
Sepal.Width	0.1251138	0.1117615

In the mean time, the formation of the two clusters could be found using function **classification** in **mclust** package: observation 1 through 41 and observation 43 though 50 are grouped in Cluster 1, and rest of the total 150 observations belong to Cluster 2.

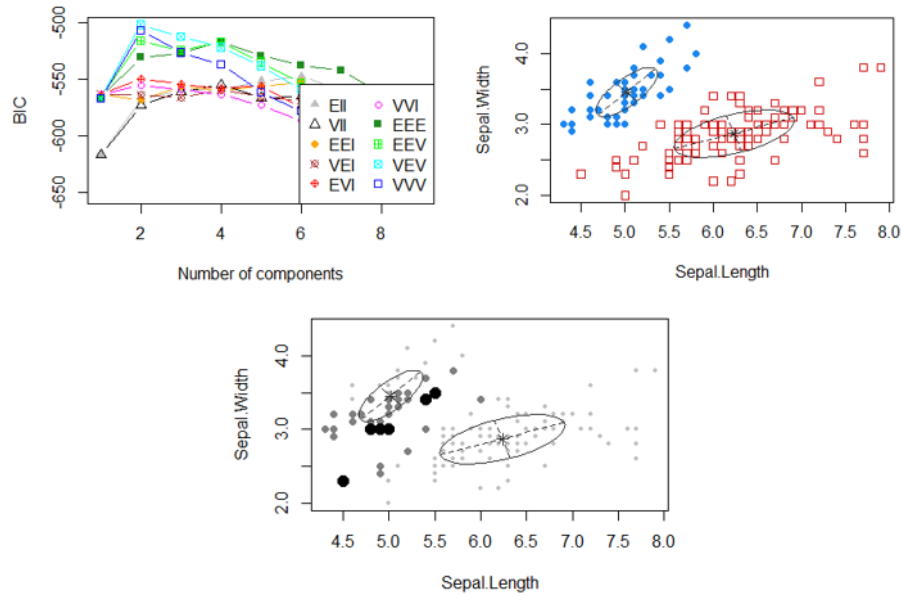


Figure 3.8: Plots Associated with the Function **Mclust** for **iris** dataset

Figure 3.8 shows three plots resulted from the **Mclust** function in the **mclust** package.

Upper left: Change of BIC from for the 10 available model parameterizations as the increase of the number of clusters up to 9. Different symbols and line types indicate different model parameterizations. The *best* model is thought of as the one with the highest BIC among the fitted models. Combined with Table 3.6, it is known that the optimal BIC -501.9589 is reached at the best number of clusters 2, by the VEV model, which specifically is a Ellipsoidal Gaussian mixture model with variable volume, equal shape and variable orientation.

Upper right: The specific clustering of **iris** data, with different symbols indicating diverse clusters corresponding to the best model as determined by function **Mclust**. The mean values for each cluster are marked and ellipses with axes are drawn corresponding to their covariances. In such case, there are two clusters, each with a different covariance.

Lower center: A projection of the **iris** data showing the uncertainty for the clustering. Larger and dark symbols indicate more uncertain observations. As is shown in the this plot, uncertain observations occur with higher frequencies at the boundaries between two clusters.

Chapter 4

Cluster Validation

4.1 Determination of Number of Clusters

Determining the number of clusters for a dataset, as illustrated in Chapter 2, is one of the two major concerns in clustering analysis, and also a frequent problem in data clustering. The quantity is often labeled k as in the K -means algorithm.

The appropriate choice of k is often fuzzy, with interpretations depending on the shape and scale of the data points' distribution and the desired clustering resolution of the user, which sometimes could be subjective. Actually, increasing k without penalty will always reduce the amount of within-cluster sum of squared distance in the resulting clustering, and in the mean time, increase the amount of between-cluster sum of squared distance, improving the percentage of explained variance for the clustering.

Under the extreme clustering situation of zero within-cluster sum of squared distance, each data point is thought of as its own cluster, then k equals the size of the data. In such case, the highest proportion of explained variance has been reached, however, leading the clustering to be meaningless, since without clustering, each original data point is its own "cluster". Therefore,

the optimal choice of k will strike a trade-off balance between a high volume of explained variance of clusters and a relatively small number of clusters. There are several categories of methods for making this decision.

4.1.1 Invalidity of Statistical Significance Testing

Firstly, one method which is commonly used but theoretically invalid for determining the number of clusters should be explained, the method is statistical significance testing, like F test based on ANOVA. Unlike many other statistical procedures, most clustering algorithms are used when there are no prior hypotheses, and cluster analysis is *structure-imposing*, sometimes it will find clusters even if none exist. Consequently, F tests are almost always misleadingly significant.

4.1.2 The Elbow Method

The elbow method looks at the proportion of variance explained as a function of the number of clusters k . An ideal value of k such that adding another cluster doesn't give much better goodness of fit of the data. In particular, if the percentage of variance explained by the clusters is drawn against the number of clusters, the first added cluster will provide much explained information (explained variance), but at some point the marginal gain will inevitably drop, giving an angle in the plot. The number of clusters is chosen right at this point, which is the "elbow criterion". This "elbow" cannot always be unambiguously identified [22].

Percentage of the explained variance is the ratio of the between-cluster variance to the total variance. A slight variation of this method plots the curvature of the within-cluster variance [23], and Figure 3.4 in Section 3.2.2 is an example of such variation of the regular elbow method. The example of the regular elbow method for determining the number of clusters, compared to Figure 3.4, could be seen in Figure 4.1, which is based on the same dataset of Romano-British Pottery [24].

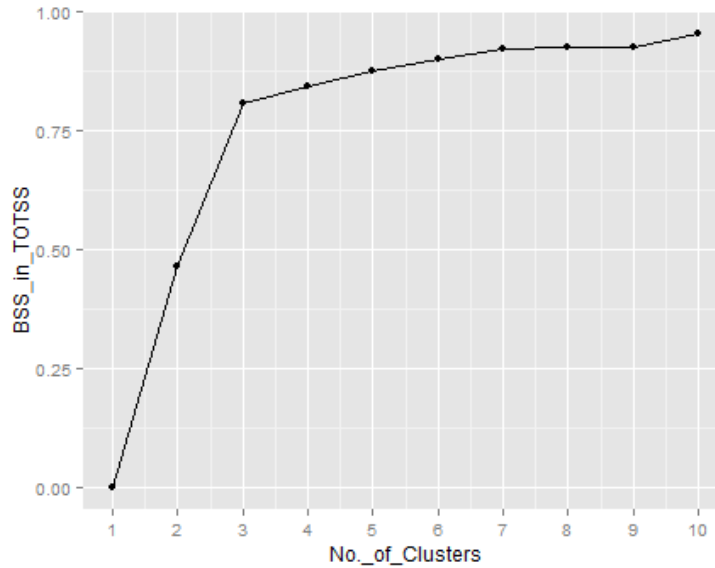


Figure 4.1: Explained Variance by Clustering against Number of Clusters

The explained variance of clustering, which is calculated as the between-cluster sum of squared distance divided by the total sum of squared distance, reaches the "elbow" when the number of clusters is three, which is the same result obtained in Figure 3.4.

At this time, 80.8 % of the total variance is explained, which is good enough and can be found in Table 4.1. When the number of clusters increases from 3 to 4, it does not add much (only 3.7%) increase in the percentage of explained variance. Thus, 3 clusters would be an ideal solution for clustering the pottery data using K -means algorithm.

Table 4.1: Percentage of Explained Variance against Number of Clusters

# C	1	2	3	4	5	6	7	8	9	10
%	≈ 0	46.6	80.8	84.5	87.5	90.2	92.4	92.7	92.6	95.5

4.1.3 Information Criterion Approach

Another set of methods for number of clusters determination are the information criteria, such as the Akaike information criterion (AIC) or Bayesian information criterion (BIC), which are often used as means for model selection in statistical modeling or predictive modeling if it is possible to construct the likelihood function for the clustering model. For example: The K -means model to some extent is a Gaussian mixture model, and the likelihood for the Gaussian mixture model is not hard to find and then also the information criterion values [25].

4.1.4 Three Top Performing Heuristic Methods

Milligan and Cooper [26] [27] developed three heuristic methods in their studies to determine the number of clusters, which are cubic clustering criterion (CCC), pseudo-F statistic and pseudo- T^2 statistic. All the three criterion or

statistic have been implemented into the statistical software SAS, and could be obtained by SAS command **PROC CLUSTER**.

- **Cubic Clustering Criterion (CCC):** the local peaks in CCC when plotted against the number of clusters provide a list of candidates for k .
- **Pseudo-F Statistic:** measures the separation among all clusters, and the local peaks in pseudo-F statistic when plotted against the number of clusters would provide a list of candidates for k .
- **Pseudo- T^2 Statistic:** measures the separation between the two clusters most recently joined, when the pseudo- T^2 is plotted against the number of clusters, the number to be one more than the peaks (or end of a run of large values) of pseudo- T^2 would provide a list of candidates for k .

Since in each of the above method, the number of local peaks may be more than one, then the three heuristic methods only provide a list of candidates for the number of clusters. In order to determine the optimal choice of k , the underlying theory of the subject being studied should be paid attention to, and other approaches for determining the k need to be used associated with the heuristic methods.

4.2 Cluster Validation

When clustering procedures are completed and the clustering results are obtained with a confirmed number of clusters and an assignment of data

points into each cluster, the next and also the final step is to evaluate the goodness of the resulting clusters, which is also known as *cluster validation*, and cluster validation usually is associated with the process of determining the number of clusters.

As for the motivation of cluster validation, it involves several concerns: to avoid finding clusters in noise, to compare different clusters, or to compare the effectiveness of different clustering algorithms on a specific dataset. One potentially useful validation technique is *Cross-validation*.

For cross-validation, firstly, randomly split the observations, and then choose one clustering technique to perform cluster analysis on each set of observations. If similar clusters develop, then such clustering result is potentially good to accept. However, if different clusters appear, then the clustering result is not generalizable. A variation on this method is to perform cluster analysis (specifically, using K -means algorithm) on the first set of observations, then use its cluster centroids as seeds to cluster the second set. This forces the same number of clusters in the cross-validation. If the cluster centroids from the first set reproduce similar assignments of data points and the clusters in the second set of observations, which have small within-cluster errors and big between-cluster errors, then this would be a good clustering.

Halkidi, et al [28] introduced the fundamental concepts of cluster validity, such as *compactness* and *separation*, and gave a systematic analysis of how cluster validity indices are used in cluster validation, including *external criteria*, *internal criteria* and *relative criteria*.

Brook, et al [29] developed an R package **clValid** which contains specific functions for validating the clustering results. There are three main types of cluster validation measures available which are "internal", "stability", and "biological", and such package can evaluate the cluster analysis resulted from up to 9 clustering algorithms, including hierarchical, K -means, self-organizing maps (SOM), model-based clustering, etc.

Appendices

Appendix A

R Code for Figure 2.2 & Clustering Analysis on Romano-British Pottery

```
## Figure 2.2 Clusters for iris Dataset on Page 6
library(devtools)
install_github('sinhrks/ggfortify')
library(ggplot2)
library(ggfortify)
library(cluster)
set.seed(1)
autoplot(fanny(iris[-5], 4), frame = TRUE)

## Clustering Analysis on Romano-British Pottery
## on Page 14-16
library(HSAUR)
kiln <- rep(1:5, c(21, 12, 2, 5, 5))
kiln <- as.data.frame(kiln)
pottery[, 10] <- kiln
pottery_dist <- dist(pottery[, colnames(pottery) !=
                             "kiln"], method = "euclidean")

# Figure 3.1: Image of Euclidean Distance based
# Dissimilarity Matrix on Pottery Data
library(lattice)
levelplot(as.matrix(pottery_dist), xlab = "Number of
                             Pot", ylab = "Number of Pot")

pottery_single <- hclust(pottery_dist, method =
                             "single")
pottery_complete <- hclust(pottery_dist, method =
                             "complete")
```

```

pottery_average <- hclust(pottery_dist , method =
                          "average")

# Table 3.2: Relations between Clusters and Kiln Sites
# for Average Link
clusters <- cutree(pottery_average , h = 4)
xtabs(~ clusters + kiln , data = pottery)

# Figure 3.2: Dendrogram of Hierarchical Clustering
# using Euclidean Distance
par(mfrow =c(1,3))
plot(pottery_single , main = "Single Link", sub = "",
      xlab = "")
plot(pottery_complete , main = "Complete Link", sub = "",
      xlab = "")
plot(pottery_average , main = "Average Link", sub = "",
      xlab = "")

```

Appendix B

R Code for *K*-means Experimental Study on Pottery Data

```
## K-means Experimental Study on Pottery Data on
## Page 17–19
library(ggplot2)
library(HSAUR)
library(HSAUR2)
set.seed(13)
res.kmeans <- lapply(1:10, function(i) {
  kmeans(pottery[, 1:9], centers = i)
})

#Within SS for each cluster (1 cluster to 10 clusters)
lapply(res.kmeans, function(x) x$withinss)

#Table 3.3: Total Within-cluster Sum of Squared Distance
res.within.ss <- sapply(res.kmeans, function(x)
  sum(x$withinss))
res.within.ss

#Figure 3.4: Relations between SSD and Number of Clusters
ggplot(data.frame(No._of_Clusters = 1:10,
  SSD = res.within.ss),
  aes(No._of_Clusters, SSD)) +
  geom_point() + geom_line() +
  scale_x_continuous(breaks = 0:10)

#Table 3.4: Cluster Means for Each Cluster &
#Table 3.5: Relations between Clusters and Kiln Sites
```

```

#using K-means
res.kmeans[3]

## Table 4.1: Percentage of Explained Variance against
## Number of Clusters on Page 35
res.between.ss <- sapply(res.kmeans, function(x)
                          (x$betweenss)/(x$totss))
res.between.ss

# Figure 4.1: Explained Variance by Clustering against
# Number of Clusters on Page 34
ggplot(data.frame(No._of_Clusters = 1:10,
                  BSS_in_TOTSS = res.between.ss),
       aes(No._of_Clusters, BSS_in_TOTSS)) +
  geom_point() + geom_line() +
  scale_x_continuous(breaks = 0:10)

```

Appendix C

R Code for Experimental Analysis on iris Dataset

```
## Experimental Analysis on Iris Dataset on Page 28–30
library(mclust)
imclust <- Mclust(iris[, 1:2])

# Table 3.6: Brief Results of EM Clustering &
# Table 3.7: Parameter Estimates of Mixing Probabilities
# and Means
# Table 3.8: Parameter Estimates of Covariances
summ <- summary(imclust, parameters = TRUE)
summ
imclust$BIC
imclust$classification

# Figure 3.6: The Bivariate iris Dataset
plot(iris$Sepal.Length, iris$Sepal.Width,
      xlab = "Sepal.Length", ylab = "Sepal.Width",
      pch = "o")

# Figure 3.7: Density Estimate for Bivariate iris Dataset
irisDens <- densityMclust(iris[, 1:2])
plot(irisDens, type = "persp", col = grey(0.8))

# Figure 3.8: Plots Associated with the Function Mclust
# for iris Dataset
plot(imclust)
```

Bibliography

- [1] Wikipedia, “Cluster analysis - wikipedia, the free encyclopedia,” 2015, [Online; accessed 15-February-2015]. [Online]. Available: http://en.wikipedia.org/wiki/Cluster_analysis
- [2] R. C. Tryon, *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Edwards brother, Incorporated, lithoprinters and publishers, 1939.
- [3] R. B. Cattell, “The description of personality: basic traits resolved into clusters.” *The journal of abnormal and social psychology*, vol. 38, no. 4, p. 476, 1943.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [5] O. Chapelle, B. Schölkopf, A. Zien *et al.*, *Semi-supervised learning*. MIT Press Cambridge, 2006.
- [6] T. Lange, M. H. Law, A. K. Jain, and J. M. Buhmann, “Learning with constrained and unlabelled data,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 731–738.

- [7] S. Basu, A. Banerjee, and R. J. Mooney, “Active semi-supervision for pairwise constrained clustering.” in *SDM*, vol. 4. SIAM, 2004, pp. 333–344.
- [8] M. Bilenko, S. Basu, and R. J. Mooney, “Integrating constraints and metric learning in semi-supervised clustering,” in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 11.
- [9] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
- [10] J. Gareth, *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013.
- [11] B. S. Everitt, “Unresolved problems in cluster analysis,” *Biometrics*, pp. 169–181, 1979.
- [12] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani, *The elements of statistical learning*. Springer, 2009, vol. 2, no. 1.
- [13] T. Hothorn and B. S. Everitt, *A handbook of statistical analyses using R*. CRC Press, 2014.
- [14] Wikipedia, “Atomic absorption spectroscopy - wikipedia, the free encyclopedia,” 2015, [Online; accessed 07-March-2015]. [Online]. Available: http://en.wikipedia.org/wiki/Atomic_absorption_spectroscopy

- [15] A. Tubb, A. Parker, and G. Nickless, “The analysis of romano-british pottery by atomic absorption spectrophotometry,” *Archaeometry*, vol. 22, no. 2, pp. 153–171, 1980.
- [16] G. H. Ball and D. J. Hall, “Isodata, a novel method of data analysis and pattern classification,” DTIC Document, Tech. Rep., 1965.
- [17] E. W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *Biometrics*, vol. 21, pp. 768–769, 1965.
- [18] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” 1973.
- [19] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.
- [20] S. Padhraic, “A guided tour of finite mixture models: From pearson to the web.” Presented as the ICML 01 Keynote Talk at Williams College, 2001.
- [21] C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca, “mclust version 4 for r: Normal mixture modeling for model-based clustering, classification, and density estimation,” 2012.
- [22] D. J. Ketchen and C. L. Shook, “The application of cluster analysis in strategic management research: an analysis and critique,” *Strategic management journal*, vol. 17, no. 6, pp. 441–458, 1996.

- [23] C. Goutte, P. Toft, E. Rostrup, F. Å. Nielsen, and L. K. Hansen, “On clustering fmri time series,” *NeuroImage*, vol. 9, no. 3, pp. 298–310, 1999.
- [24] T. Hothorn and B. S. Everitt, *A handbook of statistical analyses using R*. CRC Press, 2014.
- [25] C. Goutte, L. K. Hansen, M. G. Liptrot, and E. Rostrup, “Feature-space clustering for fmri meta-analysis,” *Human brain mapping*, vol. 13, no. 3, pp. 165–183, 2001.
- [26] G. W. Milligan and M. C. Cooper, “An examination of procedures for determining the number of clusters in a data set,” *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.
- [27] M. C. Cooper and G. W. Milligan, *The effect of measurement error on determining the number of clusters in cluster analysis*. Springer, 1988.
- [28] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On clustering validation techniques,” *Journal of Intelligent Information Systems*, vol. 17, no. 2-3, pp. 107–145, 2001.
- [29] G. Brock, V. Pihur, S. Datta, and S. Datta, “clvalid, an r package for cluster validation,” *Journal of Statistical Software (Brock et al., March 2008)*, 2011.

Vita

Lihao Zhang was born in Liaocheng, China in 1990. He received the Bachelor of Science degree in Mathematics and Applied Mathematics from Shandong University, China in 2013. He was accepted to the Master's program in Statistics in The University of Texas at Austin in 2013, and then he started his graduate studies.

Permanent address: lhzhang@utexas.edu

This report was typeset with \LaTeX^\dagger by the author.

[†] \LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's \TeX Program.